设计类脆弱性研究*)

李 艺[1] 李新明[1] 姜湘岗[2]
(装备指挥技术学院 北京101416)[1] (北京电子科技学院 北京100036)[2]

摘 要 对目前的软件脆弱性分类方法进行了分析,针对Unix/Linux操作系统,提出了基于软件脆弱性所在部件和引入原因的二维度的脆弱性分类法,并简要说明了对引入原因的进一步分类的方法。设计类脆弱性是一类重要的软件脆弱性,但在已知的各种脆弱性分类中对设计脆弱性的分类几乎是空白。本文重点对其中的设计类脆弱性进行了研究,提出了将设计类脆弱性的划分方法,分成限制脆弱性、需求无法实现脆弱性、安全设计脆弱性、异常处理脆弱性、功能局限脆弱性和随机结果脆弱性等几类,并给出了每一类设计脆弱性的定义和典型的实例。

关键词 脆弱性,分类法,操作系统,网络,软件工程

Research on Design Vulnerabilities Class

LI Yi[1] LI Xin-Ming[1] JIANG Xiang-Gang[2]
(Institution of Command and Technology of Equipment, Beijing101416)[1]
(Institute of Beijing Electronic Science and Technology, Beijing100036)[2]

Abstract This paper analyzes the existing taxonomies of software vulnerability. For Unix/Linux operation systerm, this paper proposes two-dimensional taxonomy of software vulnerability based on location and cause attributes, and describes the classification scheme of software vulnerabilities according to their cause. Design vulnerabilities class is an important class of vulnerabilites, but no existing classification identifies the types of design vulnerabilities in further detail yet. This paper elaborates on the aspect of research on design vulnerabilities and proposes the classification schema which divides design vulnerabilities into following subclasses: limit vulnerabilities, unsatisfied requirement vulnerabilities, security design vulnerabilities, exception handling vulnerabilities, restricted function vulnerabilities and random result vulnerabilities. This paper gives the definition and typical examples of every subclass.

Keywords Vulnerability, Taxonomy, Operating system, Network, Software engineering

1 概述

软件脆弱性的存在是系统安全受到威胁、系统受到攻击的根本原因。目前国内外一些大学和研究团体在脆弱性定义、脆弱性分类、脆弱性数据库的建立和维护、脆弱性扫描器、脆弱性信息共享等方面进行研究,并取得了一些进展,但成熟公认的成果有限。从软件系统的核心出发,研究脆弱性的分类,是软件脆弱性研究的一个主要内容,现在有许多不同的软件脆弱性或相关领域的分类法,但目前没有一个被普遍接受,都存在这样或那样的问题,出发点也互不相同。

我们对已知的二十几种不同的脆弱性分类方法进行研究,同时从引入原因、所在部件、直接和间接影响、修复方法、验证方法、检测方法和攻击方法等多个方面对大量典型的Linux/Unix软件脆弱性进行了深入分析,提出了基于软件脆弱性所在部件和引入原因的二维度的脆弱性分类法,将各种典型的Linux/Unix脆弱性明确地、无二义性地归类。对于软件脆弱性的引入原因分支,我们按照软件工程的观念分成设计脆弱性、编码脆弱性、配置脆弱性和环境脆弱性4类,其中设计类脆弱性在软件脆弱性中占有很大的比例,但在已知的各种脆弱性分类中对设计脆弱性的分类几乎是空白,本文对此进行了重点阐述。

2 相关的软件脆弱性分类法

软件开发是一个很复杂的过程,问题的复杂性、设计的复杂性和程序的复杂性,都会增加程序员设计和编写软件系统的难度。在需求分析和设计阶段,会由于考虑不充分、与其他系统的协调不够等原因产生各种错误,在编码阶段,各种错误和漏洞更是防不胜防,在运行阶段,会产生各种配置错误、环境错误,所以,脆弱性存在于软件生命周期的各个阶段,从需求分析、软件设计、软件编码到软件运行,每一个阶段都会引入脆弱性。

已有的软件脆弱性分类法中有一些是与引入原因相关的,最典型的是普渡大学的Aslam、krsul等提出的分类法,将脆弱性分成设计故障、代码故障、配置故障和环境故障4类,其中对代码故障和环境故障进行了细化,但设计故障一类是空白。其他的分类法如Wenliang Du等基于脆弱性生命周期的分类法中将引入原因分成输入验证错误、权限检查错误、操作序列化错误、边界检查错误、软件设计时的缺陷等几类;Kanta Jiwnani等人将脆弱性的引入原因分成有意和无意两大类,其中有意的分成恶意和非恶意两种,如特洛依木马或后门等,无意的包括有效性错误、区域错误、顺序错误、认证不完全、违反限定条件和其它逻辑错误等几类;Longstaff基于脆弱性的起

*)基金项目:863项目2003AA1Z2050.李 艺 教授,主要研究领域为操作系统和网络安全;李新明 教授,博士生导师,主要研究领域为操作系统和嵌入式系统;姜湘岗 讲师,主要研究领域为网络安全。

源将脆弱性分成缺乏训练、不允许的过程、重新引入的问题、没有推广BUG的定位、不一致的规格定义、调试代码没有删除、故障假设或判断方向错误等几类;Knuth将TEX系统的错误化分成算法偏差、补丁、数据结构错误、被忽略的函数、两个模块不匹配、对语言能力的理解错误、重新增强健壮性、超出计划外的结果、录入错误等几类;Ostrand等人提出一种属性归类方案,将错误分成数据定义错误、数据处理错误、判断错误、判断和处理错误、文档错误、系统错误和无错误几类;Basili等将软件错误分成初始化、控制结构、接口、数据和计算错误几类。

对软件脆弱性进行科学、合理、有实用价值的分类是掌握脆弱性本质属性的基础。上述这些分类法,根据对脆弱性产生原因的不同理解,针对不同的系统,采用不同归纳方法分别提出,从各自不同的角度对软件脆弱性的本质和特性进行了描述,都有其特点和可取之处,但按照分类法的原则[1]进行分析,许多都不能充分满足分类法属性必须具备的客观性、确定性、可重复性和特定性等特征,具有多义性,有许多几乎没有实用价值。

3 一个新的二维软件脆弱性分类法

针对Linux/Unix操作系统及其TCP/IP协议簇软件中的脆弱性,我们提出了基于软件脆弱性所在部件和引入原因的二维度的脆弱性分类法。其中软件脆弱性所在的部件包括操作系统、网络协议程序和其他软件。我们认为所谓软件脆弱性是指在软件的需求分析、设计、编码和运行阶段存在的漏洞,该漏洞可以在某个特定的环境被利用,从而危害系统的安全。脆弱性在软件生命周期的每一个阶段都会产生,为此,将脆弱性引入原因分成设计脆弱性、编码脆弱性、配置脆弱性和环境脆弱性4类,分类方法如表1所示。

表1

引入原因			
设计脆弱性	编码脆弱性	配置脆弱性	环境脆弱性
1)限制脆弱性 2)需求无法实现脆弱性 3)安全设计脆弱性 4)异常处理脆弱性 5)功能局限性脆弱性 6)随机结果脆弱性 7)其他设计脆弱性	1)同步互斥控制脆弱性 2)界限检查脆弱性 3)访问权限检查脆弱性 4)输入检查脆弱性 5)主体来源检查脆弱性 6)异常条件检查脆弱性 7)赋值处理脆弱性 8)其他编码脆弱性	1)安装位置脆弱性 2)安装权限脆弱性 3)安装属性脆弱性 4)环境变量配置脆弱性 5)其他配置脆弱性	

4 设计脆弱性分类

设计类脆弱性在软件脆弱性中占有很大的比例,但在已知的各种脆弱性分类中对设计脆弱性的分类几乎是空白。如普渡大学的Aslam提出的基于引入原因的脆弱性分类中,将脆弱性的引入原因分成设计错误、编码错误、配置错误和环境错误,其中对编码错误进行了详细的分类,后来krusl对环境错误部分进行了完善,但一直没有对设计错误进行进一步的完善,也没有发现其他与设计脆弱性相关的分类方法,所以我们对设计脆弱性进行了详细的分析。

设计类脆弱性是指在软件需求分析和软件设计过程中产生的软件脆弱性。由于设计类脆弱性而造成的安全漏洞不可能通过程序员正确编码、运行时正确安装、配置等措施而从根本上解决。按照软件工程的思想,软件的开发过程分成需求分析阶段、概要设计阶段、详细设计阶段、编码阶段、运行阶段和维护阶段,设计类脆弱性包括前三个阶段中出现的错误,但由于设计的粒度不同,有的错误很难确定是详细设计脆弱性还是编码脆弱性,此时要根据具体情况,分别处理。

一般来说,由设计导致的脆弱性是较难修复的,如操作系统核心中出现设计错误时,可能需要修改许多相关的模块,甚至可能需要提出新的模型。修复后可能会导致上层的应用程序不兼容等许多问题。在修复网络软件的设计错误时,如果涉及到网络交互行为,则仅仅修改自身系统是不够的,需要对方系统也进行相应的修改,而一种修复方案要得到大家的认同、成为公认的标准非常困难。

我们将设计脆弱性分成了以下几类:

4.1 限制脆弱性

指因为没有指定限制条件、指定的限定条件不充分或者指定的限制条件在现实环境下被突破而导致的脆弱性。各种软件都只有在一定的限定条件下才可以正常运行,因此在设计中要对它们的限制条件进行充分考虑,否则软件将无法正常运行,产生不可预知的执行结果。典型的实例如:千年虫、TCP SYN包湮没脆弱性、共享内存分配限额检查脆弱性等。

4.2 需求无法实现脆弱性

指由于软件中对数学或其他相关学科提出的软件需求无法真正满足而产生的脆弱性。软件在设计中,会提出各种需求,这些需求的满足可能依赖于数学或其他学科的发展,依赖于各种技术的进步。如是否有成功的随机数生成算法、加密/解密算法、散列算法等,都会影响系统需求的实现。可进一步划分为:

1)*随机数可预测脆弱性* 指由于软件使用的随机数不是真正的随机数,而是可以通过某些信息和技术进行预测的,从而导致的脆弱性。许多软件在设计中,尤其是与安全相关部分的设计中,需要获得一些"真正"的随机数,软件的成功依赖于随机数是不可预测的,而实际上,真正完全的随机数,从数学上来说是不可能的,特别是用某些固定的模式产生随机数,或者与前面的随机数有着某些联系时,后面的随机数可以被预测,从而随机数可以被伪造,造成安全漏洞。典型的实例如:TCP序列号可预测脆弱性、远程启动中事务ID脆弱性等。

2)*加密信息可破解脆弱性* 指在当前技术条件下,通过某种算法加密后的信息可以在一定时间内被破解而导致的脆弱性。所有加密信息的可靠性都是基于用户无法对加密后的内容进行破解而建立的,但由于加密算法本身的缺陷、密钥长度的限制以及解密技术的发展等,这种加密信息无法破解的需求常常无法实现。首先有许多密码破解程序可以轻松破解8字节,甚至更高字节的密码;同时由于计算机速度提高或者采用分布计算,即使采用蛮力搜索方法,也可以在较短的时间内破解密码,如使用DES Deep Crack蛮力破解工具,破解一个56位的DES密钥平均只需4.5天。一般的系统都指定一个8字节的用户密码,对用户合法性进行检查,这实际上就是一种脆

弱性，很容易被破解，如果用户选择一些不符合安全规范的密码，则问题会更严重。典型的实例如：Mac OS 弱密码加密脆弱性。

4.3 安全设计脆弱性

指在软件设计中，没有进行足够的安全考虑或者没有进行完善的安全处理而导致的各种脆弱性。此处的安全主要集中在软件已经实现或提供的功能，不包括更高安全级别软件要提供的各种安全机制，如对 C2级别的操作系统来说，没有提供 B2级别的安全机制不是脆弱性。可进一步划分为：

1）*明码信息脆弱性* 指系统或者网络传输数据中的关键信息没有加密而导致的脆弱性。系统中的关键信息在存放和缓冲的过程中，都应该加密，否则本地用户可以很容易地获得该信息。同时，由于物理线路不安全，传输的数据可能被劫持或被各种嗅探器诊听，因此在线路上传输的关键信息应该经过加密处理，否则成为安全漏洞，特别是对服务器软件而言，应该提供对各种加密信息的支持。典型的实例如：Solaris WBEM 安装脚本明文密码脆弱性、POP 密码传输脆弱性、FTP 密码传输脆弱性等。

2）*信息联想脆弱性* 指将相关联的几个信息组成在一个数据包中进行传输而导致的脆弱性。由于网络传输的不安全性，因此当信息在网络上传输被截获时，应该出于安全考虑，使得攻击者从中获得的有用信息最少。但如果某些信息包含几个部分，当该信息被截获时，可以拆解，而获得相关联的各个子信息，这就是安全漏洞。常见的如将用户名和密码，新旧密码放在一起等。信息联想和明码传输常常同时存在，如 FTP 密码传输脆弱性。这些脆弱性在许多脆弱性数据库中是作为一个脆弱性，但我们认为，采用明码传输和造成信息联想是由于不同的设计错误造成的，因此在我们的脆弱性分类方法中将这种脆弱性拆解成两个不同的脆弱性，有各自不同的修复和探测方法。典型的实例如：POP 密码脆弱性、PCMAIL 密码修改脆弱性等。

3）*信息泄漏脆弱性* 指系统信息被有意或无意泄漏而导致的脆弱性。每个系统都应该保护自身的信息，除了对本地用户开放的一些操作，对远程用户提供一些服务外，其他所有的信息都要受到系统的保护，不被非法获取或泄漏，这些信息如系统运行的操作系统和各种软件的版本信息、缓冲区中信息、网络配置信息、用户信息、文件和目录组织信息等。典型的实例如：TCP 栈指纹脆弱性、Linux netfilter NAT 信息泄漏脆弱性、I/O 系统调用文件存在脆弱性、Webmin 环境变量信息泄漏脆弱性、~ftp 目录包含系统密码文件脆弱性等。

4）*身份认证脆弱性* 指由于进行某些操作时，对调用该操作的用户或主机进行身份认证时出现错误，包括没有进行认证、认证方法有错误或者不完善等情况而导致的脆弱性。典型的实例如：SNMP 缺省字符串鉴别脆弱性、httpd 用户帐号脆弱性、FTP BOUNCE 脆弱性等。

5）*访问控制脆弱性* 指由于对某些对象（如文件、设备、邮件）进行访问时，对访问者（如主机、用户、进程）是否具有对应的访问权进行检查时出现错误，包括没有检查、检查方法错误或不完善等情况而造成的脆弱性。典型的实例如：delivermail 附加到文件脆弱性、sendmail 配置文件脆弱性、tftp 安装脆弱性、mmap 只可附加脆弱性等。

4.4 异常处理脆弱性

指因为没有考虑异常情况，或者对异常的处理不完善、不正确而导致的脆弱性。软件设计的一个关键内容就是对软件中的各种异常情况进行正确处理，一个健壮的软件必须在可能的每一个分支都正确运行，许多软件在正常的运行环境下是正确的，但在一些特殊的情况下会出现错误。典型的实例如：Linux Capabilities 脆弱性等。

4.5 功能局限脆弱性

指由于软件系统提供了不安全的功能而导致的脆弱性。软件系统提供了一些操作，这些操作是系统所必需的，或者是系统功能的一部分，但由于该操作本身含有不安全因素，因而本身就是脆弱性，很容易被攻击。这些脆弱性不可能通过自身的实现、配置等方法消除，而只能通过禁止该操作，或者通过防火墙等其他工具进行过滤才能屏蔽。典型的实例如：向外提供服务脆弱性、打开 TCP/UDP 端口脆弱性、UDP 服务拒绝脆弱性、LINUX 动态加载机制脆弱性、匿名 FTP 脆弱性、IP 碎片攻击脆弱性、rsh/rlogin 运行脆弱性、fsck 失败脆弱性、DNS 区域传输请求脆弱性、sendmail 安装 Wiz 模式脆弱性、BRU BRUEXECLOG 环境变量脆弱性等。

4.6 随机结果脆弱性

指由于软件运行时的环境不同，产生的结果不唯一而造成的脆弱性。这种结果的随机性不是软件设计本身能够检查或者处理得了的。典型的实例如：重叠（overlaping）碎片脆弱性等。

结束语 软件脆弱性产生于软件生命周期从需求分析、设计、编码到运行的各个阶段，其中由于设计缺陷而造成的软件脆弱性是不可忽视的一部分，现有的一些软件脆弱性分类法中都认识到设计类脆弱性的存在，但目前没有一个分类法对此进行了进一步分类。我们对此进行了深入研究，提出了设计类脆弱性的进一步划分方法，该方法还需要经过更多的分析和实例检验。一个好的设计脆弱性的分类，可以很好地指导我们对脆弱性的修复和探测技术的研究，同时也可以为我们研究新的、安全的、没有已知漏洞的软件设计和开发模式打下基础。

参 考 文 献

1 Krsul I V. Software Vulnerability Analysis. https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/98-09.pdf,1998

2 Krsul I. CERIAS Classic Vulnerability Database User Manual. http://ftp.cerias.purdue.edu/pub/papers/ivan-krsul/krsul-vdb-manual.pdf,2001

3 Jiwnani K, Zelkowitz M. Maintaining Software with a Security Perspective. http://citeseer.nj.nec.com/jiwnani02maintaining.html,2002

4 Ritchey R,O'Berry B,Noel S. Representing TCP/IP Connectivity For Topological Analysis of Network Security. http://www.acsac.org/2002/papers/73.pdf,2002

5 Wilander J,Kamkar M. A Comparison of Publicly Available Tools for Dynamic Buffer Overflow Prevention. http://citeseer.nj.nec.com/wilander03comparison.html,2003

6 Tsai T,Singh N. Libsafe 2.0: Dectection of Format String Vulnerability Exploits,2001